

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

«На правах рукопису»

УДК _____

До захисту допущено:

Завідувач кафедри

_____ Сергій Стіренко
“ ____ ” _____ 2020 р.

Магістерська дисертація

на здобуття ступеня магістра

за освітньо-науковою програмою «Комп’ютерні системи та мережі»

зі спеціальності 123 «Комп’ютерна інженерія»

на тему: «Засоби створення мобільного додатку Click’n go»

Виконав (-ла):

студент (-ка) VI курсу, групи ІО-391мп

Клименко Ярослав Володимирович _____

Керівник:

Доцент кафедри ОТ, к.т.н.

Верба Олександр Андрійович _____

Консультант з нормоконтролю:

Професор кафедри ОТ, д.т.н., проф.

Кулаков Юрій Олексійович _____

Рецензент:

Доцент кафедри СП і СКС, к.т.н., доц.

Орлова Марія Миколаївна _____

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.

Студент _____
(підпис)

Київ – 2020 року

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМ. І.СІКОРСЬКОГО»**

**ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ
КАФЕДРА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ**

Рівень вищої освіти – другий (магістерський)
Напрямок підготовки **123** – «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ
Завідувач кафедри

(підпис) С.Г. Стіренко
(ініціали, прізвище)

" ____ " _____ 2020 року

**ЗАВДАННЯ
НА МАГІСТЕРСЬКУ ДИСЕРТАЦІЮ СТУДЕНТА**

Клименко Ярослава Володимировича
(прізвище, ім'я, по батькові)

1. Тема проекту Засоби створення мобільного додатку Click`n go
керівник проекту Доцент кафедри ОТ, к.т.н., Вербка О. А.

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «26» жовтня 2020 року № 3133-с

2. Строк подання студентом дисертації 26.11.20

3. Вихідні дані до проекту – дивись технічне завдання

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити)
Огляд існуючих рішень: загальні відомості про туристичних гідів, розгляд прикладів
готових реалізацій. Проектування додатку: опис предметної області, визначення
вимог і завдань, функції додатку, прецеденти, структура додатку. Розробка додатку:
вибір технологій та його обґрунтування, основні рішення з реалізації додатку та

його компонентів. Висновки.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
Схема роботи додатку(1 шт), Схема ресурсів додатку(1 шт),

6. Консультанти розділів проекту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1-3	Кулаков Ю.О., проф. каф. ОТ, д.т.н.		

7. Дата видачі завдання 01.08.2019

КАЛЕНДАРНИЙ ПЛАН

N з/п	Назва етапів дипломного проекту	Строк виконання етапів проекту	Примітка
1	Вивчення необхідної літератури	20.08.2019	
2	Огляд існуючих рішень	15.09.19	
3	Складання і узгодження технічного завдання	01.10.2019	
4	Написання вступної частини та огляд рішень	12.02.2020	
5	Розробка архітектури додатку	01.03.2020	
6	Тестування та виправлення помилок	08.03.2020	
7	Оформлення документації дипломного проекту	18.04.2020	

Студент _____
 (підпис)

Клименко Я.В.
 (прізвище та ініціали)

Керівник проекту _____
 (підпис)

Верба О.А.
 (прізвище та ініціали)

Реферат

Магістерська дисертація: 69 с., 20 рис., 30 табл., 3 додатки, 17 джерел.

Актуальність проблеми. Відсутність на ринку швидкого та зручного рішення проблеми пошуку закладу на місцевості.

Зв'язок роботи з науковими програмами, планами, темами. Є авторською роботою у сфері веб-розробки.

Мета і задачі дослідження. Завданням цієї роботи є розробка більш продуктивного методу пошуку закладів та подальша розробка додатку з використанням розробленого підходу з технологіями веб-розробки. Метою є розв'язання задач та отримання програмного продукту, що спрямований на подальше використання користувача.

Об'єкт дослідження. Об'єктом дослідження є інтеграція методів веб-розробки.

Предмет дослідження. Додаток для генерації закладу на місцевості.

Новизна одержаних результатів. Отримання програмного продукту, що спрямований на економію часу на знаходження найближчих закладів, на відміну від більшості додатків з подібним стеком технологій, що спрямовані у більшості на елемент соціальної мережі.

Ключові слова. *Веб-розробка, геокодування, API, auth, JavaScript.*

Abstract

Master's dissertation: 69 pages, 20 figures, 30 tables, 3 appendices, 17 sources.

The urgency of the problem. Lack of a quick and convenient solution to the problem of finding an institution in the area.

Relationship of work with scientific programs, plans, themes. Is an author's work in the field of web development.

Purpose and tasks of the research. The task of this work is to develop a more productive method of finding institutions and further development of the application using the developed approach with web development technologies. The goal is to solve problems and obtain a software product that is intended for further use by the user.

Object of study. The object of research is the integration of web development methods.

Subject of study. An application for generating an institution on the ground.

The novelty of the results. Obtaining a software product that aims to save time to find the nearest institutions, in contrast to most applications with a similar stack of technologies, which are mostly aimed at the element of the social network.

Keywords. *Web development, geocoding, API, auth, JavaScript.*

АНОТАЦІЯ

Даний магістерський дипломний проект присвячений дослідженню методів розробки веб-додатку. Метою розробка більш продуктивного методу пошуку закладів та подальша розробка додатку з використанням розробленого підходу з технологіями веб-розробки.

У роботі проведено дослідження готових реалізацій наближених до системи програм, визначено їх переваги та недоліки та розроблено вимоги до даного додатку.

Розроблений програмний додаток дозволяє користувачеві у режимі реального часу задавати параметри пошуку закладу і знайти найближчий на карті. Програма реалізована за допомогою засобів мови програмування JavaScript Для реалізації функцій додатку було використано бібліотеку React.

ANNOTATION

This master's thesis project is devoted to the study of web application development methods. The purpose of the thesis is to develop an application for convenient and fast search of the institution in the area.

The study of ready implementations of programs close to the system is carried out, their advantages and disadvantages are determined and the requirements to this appendix are developed.

The developed software application allows the user to set the search parameters of the institution in real time and find the nearest one on the map. The program is implemented using the JavaScript programming language. The React library was used to implement the functions of the application.

Технічне завдання до магістерської дисертації

на тему: «Засоби створення мобільного додатку Click`n go»

Київ – 2020

ЗМІСТ

1.	НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ.....	2
2.	ПІДСТАВИ ДЛЯ РОЗРОБКИ.....	2
3.	МЕТА ТА ПРИЗНАЧЕННЯ РОЗРОБКИ.....	2
4.	ДЖЕРЕЛА РОЗРОБКИ.....	2
5.	ТЕХНІЧНІ ВИМОГИ.....	3
6.	ЕТАПИ РОЗРОБКИ.....	4

1. НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ

Технічне завдання стосується розробки додатку для реалізації швидкого та зручного рішення для пошуку закладу на місцевості.

Областю застосування системи є генерація закладу та пошуку її на карті.

2. ПІДСТАВИ ДЛЯ РОЗРОБКИ

Підставою для розробки є завдання на виконання магістерської дисертації, затверджене кафедрою обчислювальної техніки Національного технічного університету України «Київський політехнічний інститут ім. І.Сікорського».

3. МЕТА ТА ПРИЗНАЧЕННЯ РОЗРОБКИ

Метою є дослідження засобів створення більш продуктивного пошуку закладів на місцевості та подальша розробка додатку з використанням розробленого підходу з технологіями .

4. ДЖЕРЕЛА РОЗРОБКИ

Джерелом розробки є науково-технічна література, публікації в спеціалізованих періодичних виданнях, довідники по платформах дистанційного навчання, публікації в мережі Інтернет по даній темі.

5. ТЕХНІЧНІ ВИМОГИ

Додаток, що розробляється повинен:

- Надавати можливість задання параметрів пошуку закладу;
- Генерація закладу по заданим параметрам;
- Відображати на карті місцезнаходження закладу від заданої точки.

6. ЕТАПИ РОЗРОБКИ

Назва етапу	Дата
Вивчення необхідної літератури	20.08.2019
Огляд існуючих рішень	15.09.19
Складання і узгодження технічного завдання	01.10.2019
Написання вступної частини та огляд рішень	12.02.2020
Розробка архітектури додатку	01.03.2020
Тестування та виправлення помилок	08.03.2020
Оформлення документації дипломного проекту	18.04.2020

**Пояснювальна записка
до магістерської дисертації**

на тему: «Засоби створення мобільного додатку Click`n go»

Київ – 2020

ЗМІСТ

ПЕРЕЛІК ТЕРМІНІВ ТА СКОРОЧЕНЬ	7
ВСТУП	8
РОЗДІЛ 1	10
ОГЛЯД ІСНУЮЧИХ РІШЕНЬ	10
1.1. Загальні відомості про мобільні гіди.	10
1.2. Огляд існуючих рішень	11
1.2.1. Foursquare.....	11
1.2.2. Yelp.....	13
1.2.3. Tripadvisor.....	14
ВИСНОВОК ДО РОЗДІЛУ 1	17
РОЗДІЛ 2	18
ПРОЕКТУВАННЯ ДОДАТКУ	18
2.1. Опис предметної області	18
2.1.1. Автентифікація.....	18
2.1.2. Геокодування.....	18
2.2. Визначення вимог і завдань	19
2.3. Опис функціоналу додатку	19
2.4. Дизайн та сценарій.....	20
2.4.1. Завантажувальний екран	20
2.4.2. Реєстрація та вхід.....	21
2.4.3. Сторінка з вибором сегменту закладу	23

2.4.4. Генерація закладу	26
2.4.3. Пошук закладу на карті	27
2.5. Прецеденти	28
2.5.1. Обрати сегмент закладу	29
2.5.2. Обрати кількість людей.....	29
2.5.3. Обрати середню вартість меню на одну людину.....	31
2.5.4. Знайти згенерований заклад на карті.....	32
2.6. Архітектура серверної частини	33
ВИСНОВОК ДО РОЗДІЛУ 2	36
РОЗДІЛ 3	37
РОЗРОБКА ДОДАТКУ	37
3.1. Обґрунтування вибору технологій.....	37
3.1.1. Вибір платформи для додатку	37
3.1.2. Вибір мови програмування	38
3.1.3. Вибір допоміжних бібліотек та фреймворків	38
3.2 Рішення обміну даними.....	44
3.2.1 API	44
3.2.1 Форма обміну даними.....	44
3.2.3. Транспортування даних.....	45
3.3. База даних	45
3.4. Сервер.....	46
3.4.1 Веб-сервер.....	46
3.4.2 Віртуальний виділений сервер	47

ВИСНОВОК ДО РОЗДІЛУ 3	48
РОЗДІЛ 4	50
РОЗРОБКА СТАРТАП-ПРОЕКТУ	50
4.1. Опис ідеї проекту	50
4.2. Технологічний аудит ідеї проекту.....	51
4.3. Аналіз ринкового потенціалу для старту стартап-проекту	52
4.4. Розробка ринкової стратегії стартап-проекту	58
4.5. Розроблення маркетингової програми стартап-проекту.....	60
ВИСНОВОК ДО РОЗДІЛУ 4	64
ВИСНОВКИ.....	66

ПЕРЕЛІК ТЕРМІНІВ ТА СКОРОЧЕНЬ

ОС	Операційна система.
SDK	(англ. Software development kit) – набір із засобів розробки, утиліт і документації, який дозволяє програмістам створювати прикладні програми за визначеною технологією або для певної платформи.
Фреймворк	(англ. Framework) – Програмне забезпечення, що полегшує розробку складних систем.
API	Прикладний програмний інтерфейс
GPS	Система глобального позиціонування
UI	Інтерфейс користувача
User-friendly	Інтуїтивно зрозумілий користувачу
HTTP/HTTPS	Інтерфейс користувача
JSON	формат даних
URL	Система уніфікованих адрес електронних ресурсів
VDS	Віртуальний виділений сервер

ВСТУП

Актуальність дослідження. На сьогоднішній день, складно знайти людину, яка не має мобільного пристрою та ні разу не відвідувала інші країни або хоча б міста. Люди подорожують з різних причин - це може бути відпустка, або ж відрядження по роботі. І звичайно ж, в таких мандрівках мобільний пристрій завжди під рукою, який в свою чергу дає можливість швидко знайти будь-яку інформацію за бажанням.

І саме тому, в наш час, електронний довідник мандрівок - має великий попит серед користувачів-туристів. В свою чергу, такі довідники повинні бути зручними, швидкими та взагалі «юзерфрендлі». Адже в сучасному ритмі життя, користувач не хоче витратити лишній час на довгі планування та не зрозумілий інтерфейс.

Хорошими прикладом таких електронних довідників є – TripAdvisor, який ми будемо детально розглядати в 1 розділі даної дисертації. Основною ідеєю такого електронного довідника - є те що , користувач з легкістю, де б він не перебував, маючи лише смартфон підключений до інтернету та встановлений додаток за допомогою лише декількох дотиків - дізнатись де він може переночувати, знайти їжу або просто провести певний час з інтересом. Також він може дізнатись від інших користувачів, які залишали коментарі та відгуки, інформацію про те, щодо закладу, який збирається відвідати - його переваги та недоліки перед іншими закладами, або ж залишити після відвідування закладу свій коментар та відгук, який в свою чергу допоможе іншому туристу витрачаючи мінімум часу обрати саме той заклад, який буде до вподоби саме йому.

Метою даної роботи є дослідження засобів створення мобільного додатку, який використовує вподобання користувача та радить місце для відвідування.

Відповідно з метою були поставленні такі завдання:

1. Розглянути готові рішення та виділити переваги і недоліки .
2. Аналіз мов програмування для написання мобільного додатку.
3. Аналіз корисних API для написання додатку.
4. Аналіз ринку користувачів.
5. Розробка алгоритму роботи мобільного додатку.
6. Створення мобільного додатку.

Об’єктом дослідження являється дослідження засобів розробки мобільного додатку.

Предметом дослідження являється додаток «електронний довідник мандрівника».

РОЗДІЛ 1

ОГЛЯД ІСНУЮЧИХ РІШЕНЬ

1.1. Загальні відомості про мобільні гід

Представлені додатки в цьому розділі є областю веб та мобільної розробки.

Ціллю та основною ідеєю таких додатків є полегшити життя людей, а саме життя туристів та мандрівників, які фізично не можуть знати про розташування, популярність або якість обслуговування в тому або іншому закладі не знайомого міста або країни, також такі додатки вносять певну цікавинку та розвагу (приклади будуть наведені далі у розділі).

На сьогоднішній день немає ідеального додатку, який би сподобався всім користувачам своїм функціоналом або гарантією достовірної та об'єктивної інформацією.

Основною функцією додатку являється представлення найближчих закладів від вашого розташування для того, щоб все працювало правильно в телефоні повинен бути GPS, а прикладним терміном який пояснює, як все працює (представлення гео-даних в вигляді адреси або коду вулиці тощо.) є Геокодування – який ми розглянемо і ознайомимся ближче в 2 розділі цієї роботи.

Також основною функцією являються елементи соціальної мережі (Залишити коментар або відгук, поставити оцінку тощо), адже користувач повинен мати змогу залишити свій відгук після отриманих емоцій від відвідування того чи іншого закладу для того, щоб надалі формувався спільний рейтинг закладів в кожному сегменті.

1.2. Огляд існуючих рішень

1.2.1. Foursquare

Foursquare (4sq) - це програма для відмітки геолокації, призначена в основному для роботи з мобільними пристроями. Ця послуга доступна для користувачів не тільки з пристроями, оснащеними GPS-навігацією, наприклад, для користувачів смартфонів, але і для роботи з будь-яким мобільним телефоном. Якщо мобільний телефон не оснащений GPS-навігацією, місцезнаходження визначається LBS (Location Based Service). Користувачі мають позначку ("реєстрація") у різних закладах, використовуючи мобільну версію сайту, SMS або спеціальний додаток, призначений для конкретної операційної системи мобільного пристрою. Кожен з цих рейтингів дозволяє користувачеві заробляти чотири квадратні бали. З травня 2014 року можливість "zareєstrуватися" в установах за допомогою Foursquare була вилучена та перенесена в окремий додаток: Swarm. З тих пір сервіс позиціонував себе як додаток, що рекомендує мандрівникам, де їх можна відвідати (наприклад, TripAdvisor, Yelp). Рекомендації формуються на основі уподобань користувачів, рейтингу місцеположень та пропозицій, залишених іншими користувачами Foursquare. У малому. 1.1 показаний приклад цієї програми.

Переваги:

- 1) Система "Checkin" , яка допомагає побачити де були ваші друзі, також за "Checkin" в незвичному місці (наприклад космічна станція) – особливі помітки.
- 2) Величезна база даних місць по всьому світу, яка постійно оновлюється.

- 3) Ігрова складова програми (в вигляді «Check in»), яка захоплює мільйони користувачів.
- 4) Інтеграція з іншими соціальними мережами.

Недоліки:

- 1) Проблеми з локалізацією місць на карті.
- 2) Неможливість зміни мови додатка.
- 3) Проблеми з установкою з'єднання, нестабільність в роботі.
- 4) Наявність в базі дублікатів місць і місць сумнівного походження.

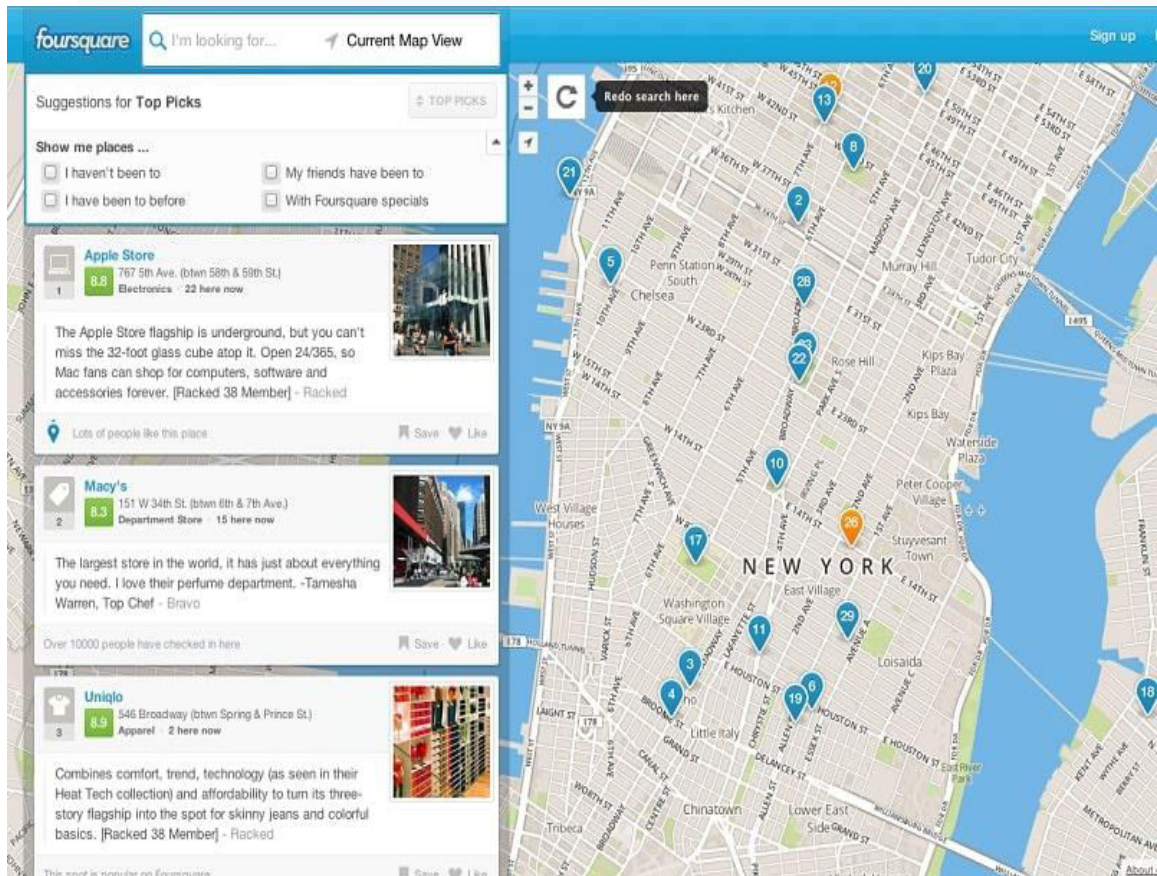


Рисунок 1.1 – Приклад роботи Foursquare

1.2.2. Yelp

Yelp - сайт для пошуку послуг на місцевості, наприклад ресторанів або барберів, з можливістю додавати та переглядати рейтинги та огляди цих послуг. Є сотні оглядів для популярних місць. Для користувачів сайту були зроблені елементи соціальної мережі. Сервіс був запущений в жовтні 2004 року в Сан-Франциско, в подальшому географія була розширена. Найбільшим ринком компанії залишаються США, а всього компанія Yelp створила для понад 111 країн (2013) сайти послуг. Відвідуваність сайтів Yelp на кінець 2010 року була понад 39 млн унікальних відвідувачів що місяця, до початку 2016 року - більше 142 млн унікальних відвідувачів що місяця. На рис 1.2 зображено приклад роботи додатку та сайту.

Переваги:

- 1) Перевірена (надійна) база закладів та відгуків про заклади.
- 2) Багатофункціональність.

Недоліки:

- 1) Застарілий дизайн.
- 2) Дуже довгі запити до серверу.

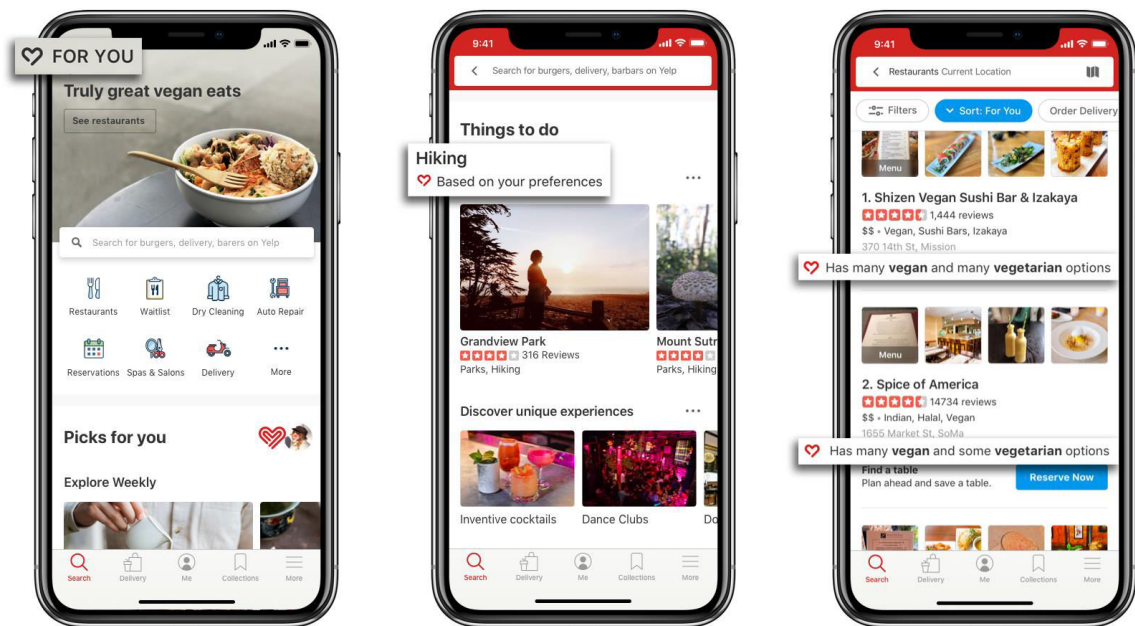


Рисунок 1.2 – Приклад роботи Yelp

1.2.3. Tripadvisor

TripAdvisor — веб-сайт, для користувачів які планують відвідати будь-яку країну в майбутньому. Послуги сайту є безкоштовними для користувачів, які активно добавляють контент. За даними офіційного веб-сайту, додаток та сайт TripAdvisor щомісяця має 315 мільйонів унікальних відвідувачів, та більше 70 мільйонів зареєстрованих користувачів, які залишили понад 200 мільйонів відгуків про заклади відпочинку і не тільки. На основі рейтингу користувачів TripAdvisor формує списки самих популярних місць. Чому це так популярно? Допустимо, ви зацікавитесь якою-небудь країною або містом і не знаєте, що там є цікаве. Або ви вже склали маршрут, але у вас ще є вільний час, який теж хотілося провести цікаво. Додаток TripAdvisor дає можливість без великих часових затрат на вивчення довідників та

прочитання довгих форумів, які можна переглянути, чим може сподобатись місто чи країна, які місця частіше відвідують туристи. Основною перевагою TripAdvisor є алгоритм, який відбирає серед великої купи відгуків фейкові, таким чином розробник заявляє, що всі відгуки на заклади були написані людьми які дійсно там були.

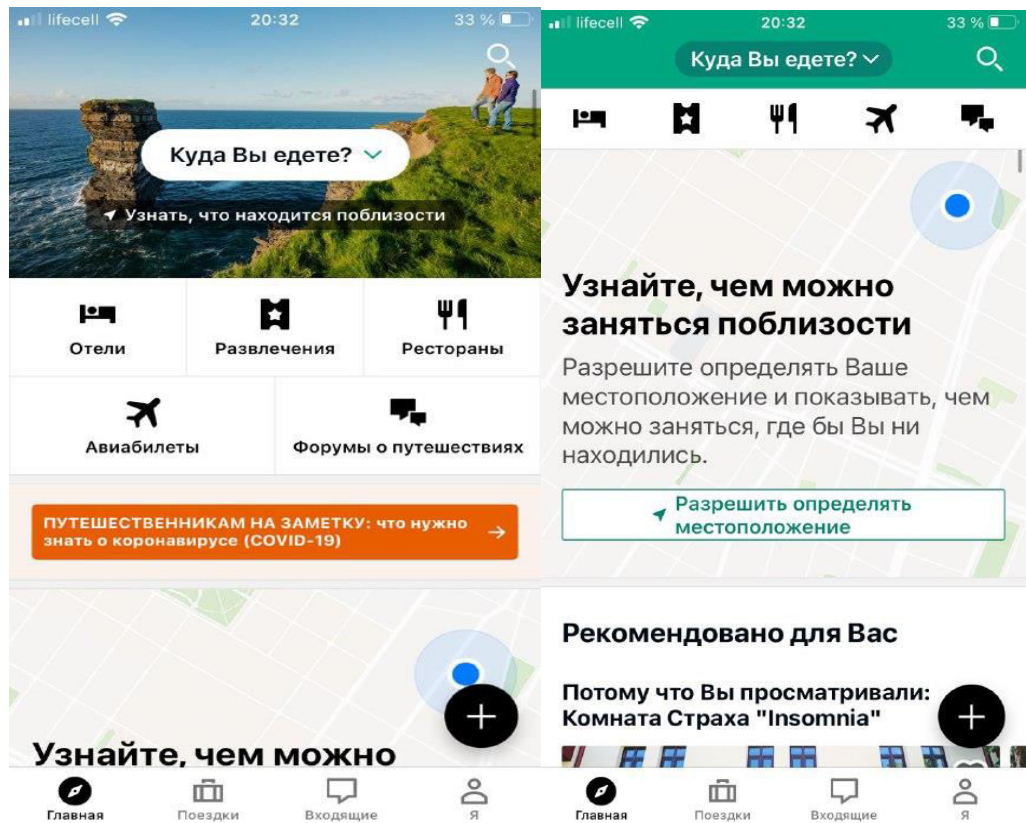


Рисунок 1.3 – Приклад роботи додатку TripAdvisor

Переваги:

- 1) Чітка система відстеження відгуків.
- 2) Зручний дизайн.

Недоліки:

- 1) Оскільки рейтинг закладу формується на підставі оцінювання людей, які відвідували його, запропоновані заклади можуть не сподобатись користувачу.

ВИСНОВОК ДО РОЗДІЛУ 1

Під час виконання роботи було проведено дослідження готових реалізацій схожих додатків «гідів», пошуку місця для відпочинку та проведення часу. Представлені найбільш популярні на сьогоднішній день рішення, які працюють під різними мобільними ОС. Кожна з оглянутих реалізацій має ряд переваг та недоліків, провівши аналіз яких можна сформулювати оптимальні вимоги до функціоналу розроблюваного застосунку для задоволення основних потреб користувача.

Розглянемо основні недоліки готових рішень:

- 1) Дуже багато функціоналу іноді хочеться якомога швидше знайти потрібний заклад без зайвих дій.
- 2) Велика кількість не правдивих відгуків та оцінок які формувались на основі суб'єктивних вражень відвідувачів, це може стати проблемою для системи пошуку потрібного закладу.

На основі проаналізованих недоліків складемо основні вимоги до розроблювального додатку:

- 1) Додаток повинен бути максимально простий та мати дружелюбний інтерфейс юзера.
- 2) Генерація максимально підходящого згідно вимог користувача місця для відпочинку та проведення часу.
- 3) Мінімалістичний та сучасний дизайн.

Для усунення зазначених недоліків існуючих аналогічних реалізацій, необхідно розробити застосунок із зазначеними вище вимогами.

РОЗДІЛ 2

ПРОЕКТУВАННЯ ДОДАТКУ

2.1. Опис предметної області

2.1.1. Автентифікація

Перед тим як користуватись додатком, користувач повинен пройти форму реєстрації, тому доцільно буде розглянути методи рішення задачі автентифікації.

Автентифікація — це перевірка на унікальність пред'явленого користувачем ідентифікатора. Основним передбаченим результатом автентифікації є авторизація користувача, тобто надання йому прав доступу до ресурсів до яких у нього є права, для виконання його завдань. Залежно від важливості та безпеки ресурсу, для доступу до нього можуть використовуватися різні методи автентифікації. Для рішення проблеми автентифікації буде доцільно використати бібліотеку Passport.js , яку більш детальніше розглянемо в розділі 3.

2.1.2. Геокодування

Ще однією з поставлених задач є реалізація карт з правильними геоданами закладів.

Геокодування (*Geocoding*) — призначення місцю універсального географічного ідентифікатора, яке задається у вигляді поштової адреси або унікальною назвою. Зворотнє геокодування використовує географічні координати (точки на земній кулі у вигляді широти і довготи), щоб знайти опис місця (поштову адресу, назву поселення і т.д.).

Геоінформаційна система — сучасна комп'ютерна технологія, що поєднує модельне зображення території (електронне відображення карт, схем, зображення земної поверхні з повітря або космосу) з інформацією у таблиці такі як статистичні дані, списки, економічні показники . Також, під геоінформаційною системою розуміють так звану систему управління просторовими даними та асоційованими з ними атрибутами. Конкретніше, це комп'ютерна система, що забезпечує можливість використання, збереження, редагування, аналізу та відображення географічних даних. Для реалізації підключення карт до додатку буде доречно використати готовий API від Google який називається Google Maps JavaScript API V3.

2.2. Визначення вимог і завдань

Основними функціями додатку є:

- 1) Генерування закладу на основі вхідних даних в вигляді обраних уподобань користувача.
- 2) Пошук найближчих і відповідних до обраних параметрів закладів.

Основні вимоги до додатку:

- 1) Зрозумілий графічний інтерфейс;
- 2) Охоплення широкого кола потенційних користувачів.
- 3) Швидкість у роботі .

2.3. Опис функціоналу додатку

Додаток повинен надавати користувачу можливість здійснювати наступні маніпуляції:

- 1) Обрати сегмент закладу для пошуку найближчих або найліпших за вподобаннями користувача (всього виділяється 5 основних сегментів):

- a. Кафе (заклади де можна випити кави або чаю);
 - b. Їжа (заклади де можна знайти їжу);
 - c. Веган (заклади де можна знайти веганську їжу);
 - d. Бар (заклад де можна знайти спиртні напої та котейлі);
 - e. Кальян-бар (заклади де можна знайти кальян);
- 2) Обрати кількість людей для яких надалі буде обраний заклад:
- a. Для одного
 - b. Для двох
 - c. Для дружньої компанії (від 3 до 5 осіб)
 - d. Для великої компанії (від 6 до 10 осіб)
 - e. Для корпоративу або вечірки (більше 10 осіб)
- 3) Обрати середню суму на людину яку має користувач
- a. Економний варіант
 - b. Середня вартість
 - c. Велика вартість
 - d. Бізнес
 - e. Люкс
- 4) Обрати місце де ви знаходитесь та згенерувати заклад

2.4. Дизайн та сценарій

Після узагальнення всіх кореневих функцій, котрі мусять бути реалізовані у додатку, сформуємо сценарії використання додатку та його наближений дизайн за допомогою створеного макету майбутнього додатку.

2.4.1. Завантажувальний екран

На рис. 2.1 зображено завантажувальний екран програми який буде зустрічати користувача кожен раз при запуску програми.



Рисунок 2.1 – Приклад завантажувального екрану додатку

2.4.2. Реєстрація та вхід

На рис. 2.2 зображено екран Реєстрації та входу в додаток. Користувачу буде запропоновано увійти в додаток, якщо він має обліковий запис або створити новий якщо не має.

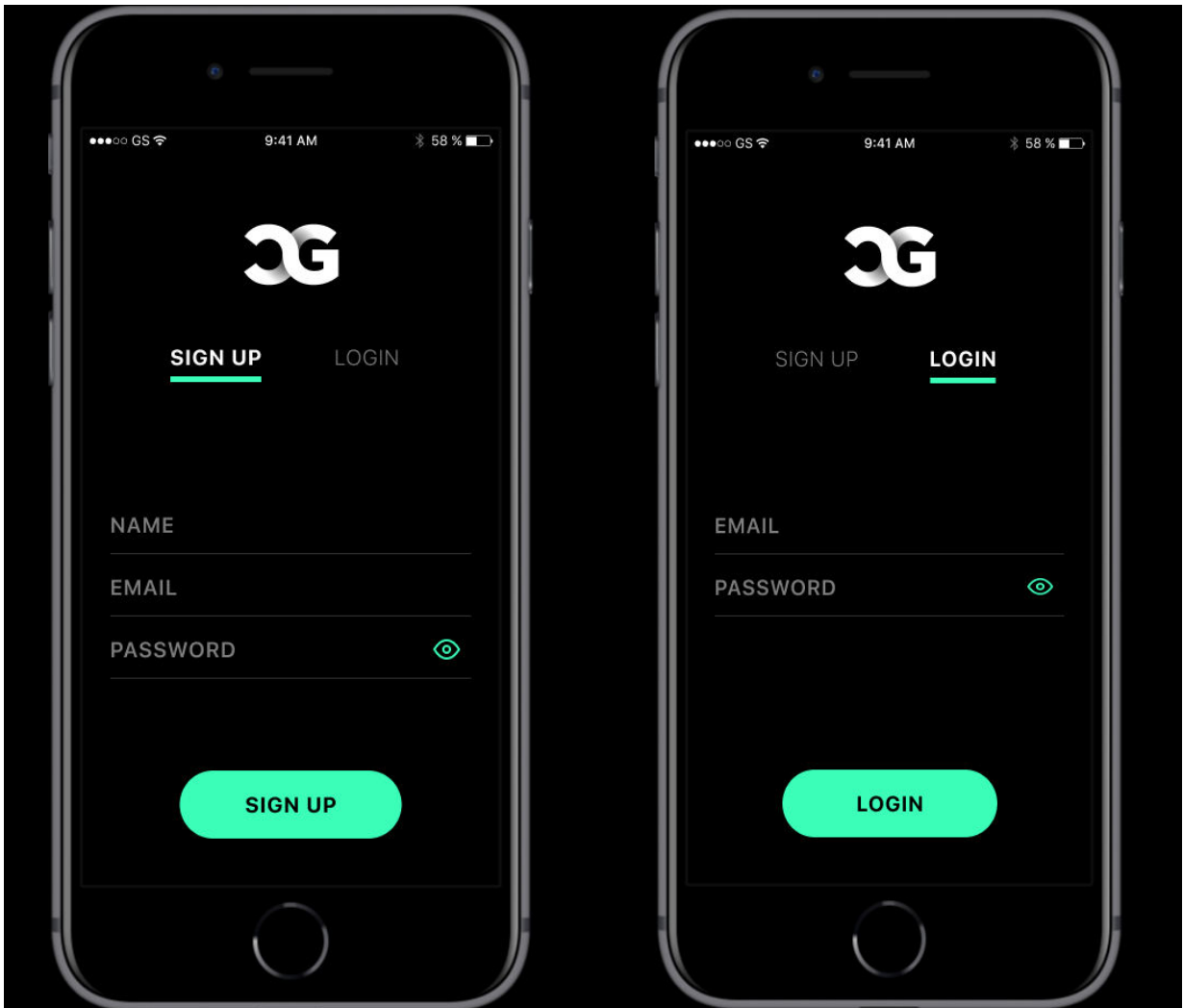


Рисунок 2.2 – Екран реєстрації

Після того як користувач ввів дані його зустрічає знову завантажувальний екран рис. 2.3.

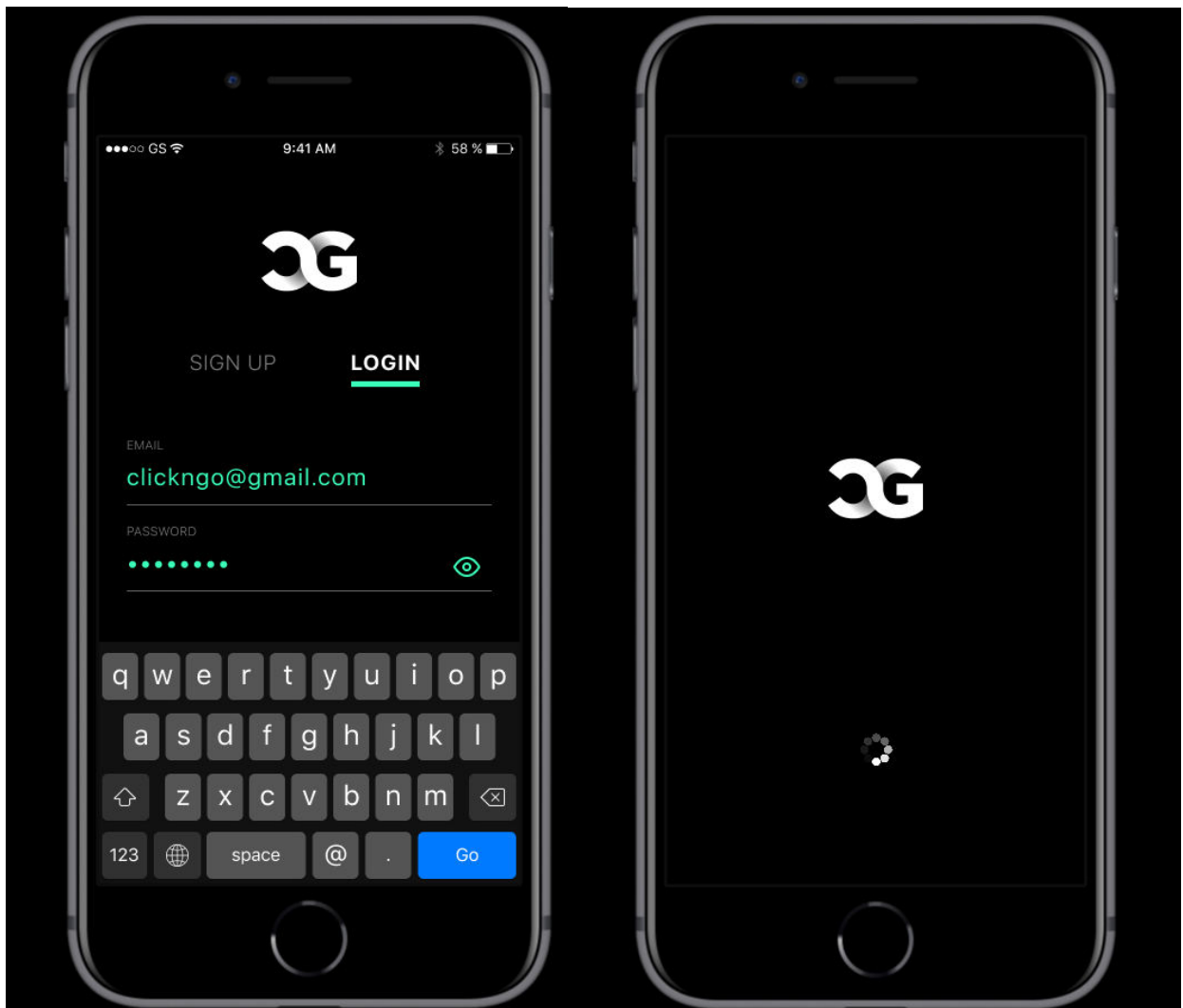


Рисунок 2.3 – Экран входу

2.4.3. Сторінка з вибором сегменту закладу

Після виконаного входу користувача зустрічає основний інтерфейс додатку, а саме сторінка з вибором сегменту закладу, кількості людей, середньої ціни меню, зображеної на рис. 2.4, 2.5, 2.6 відповідно.

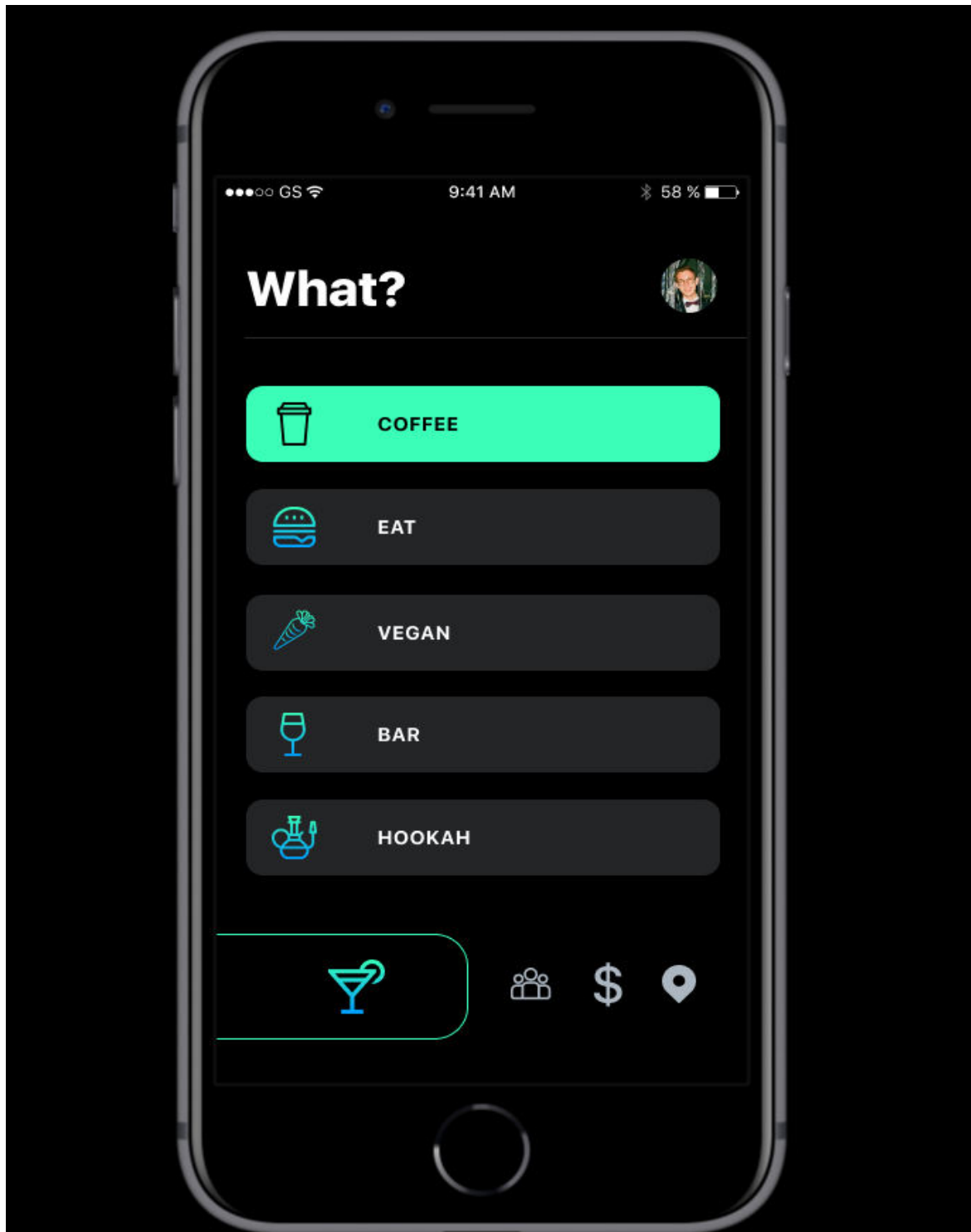


Рисунок 2.5 – Вибір сегменту закладу та кількості

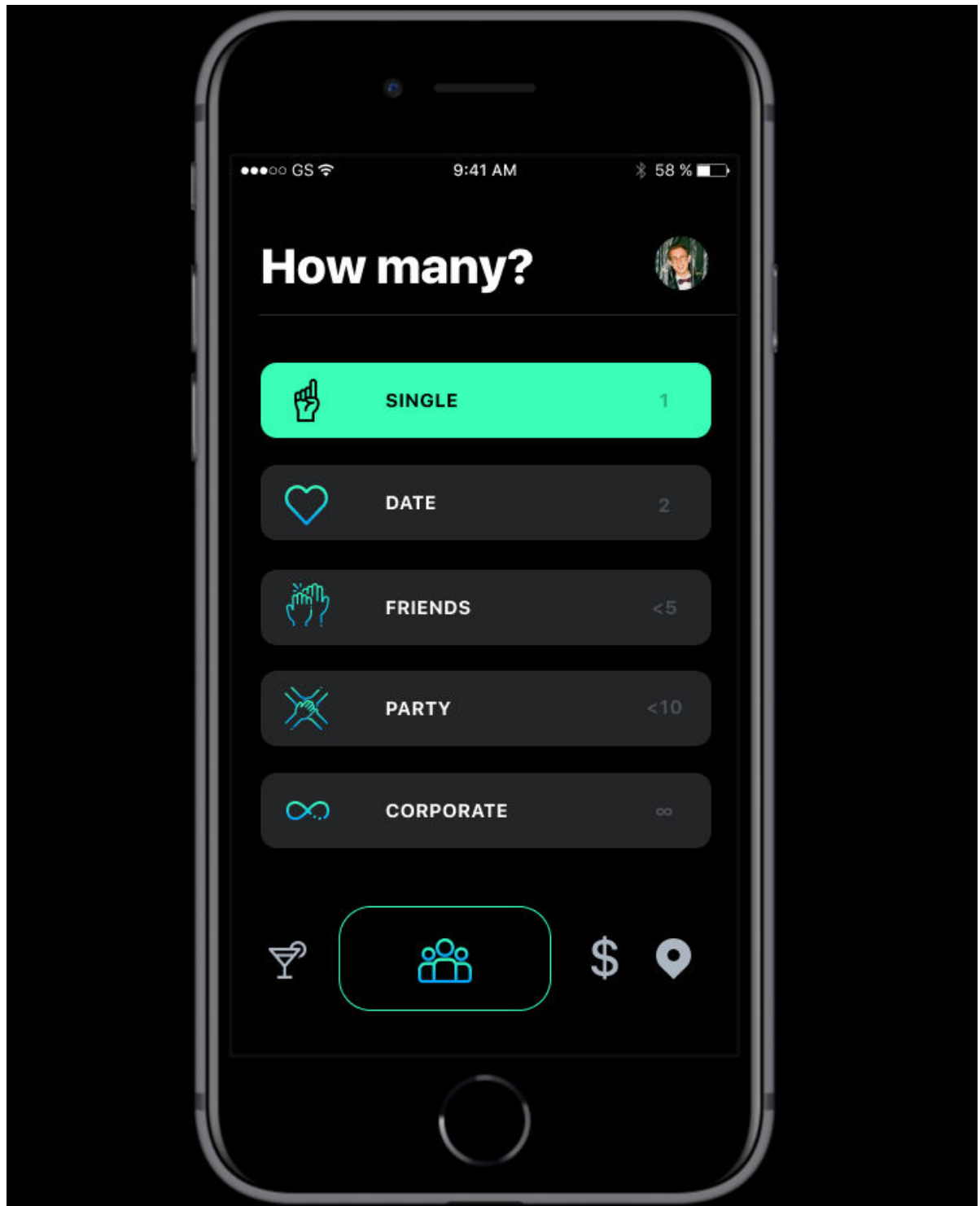


Рисунок 2.6 – Вибір кількості людей

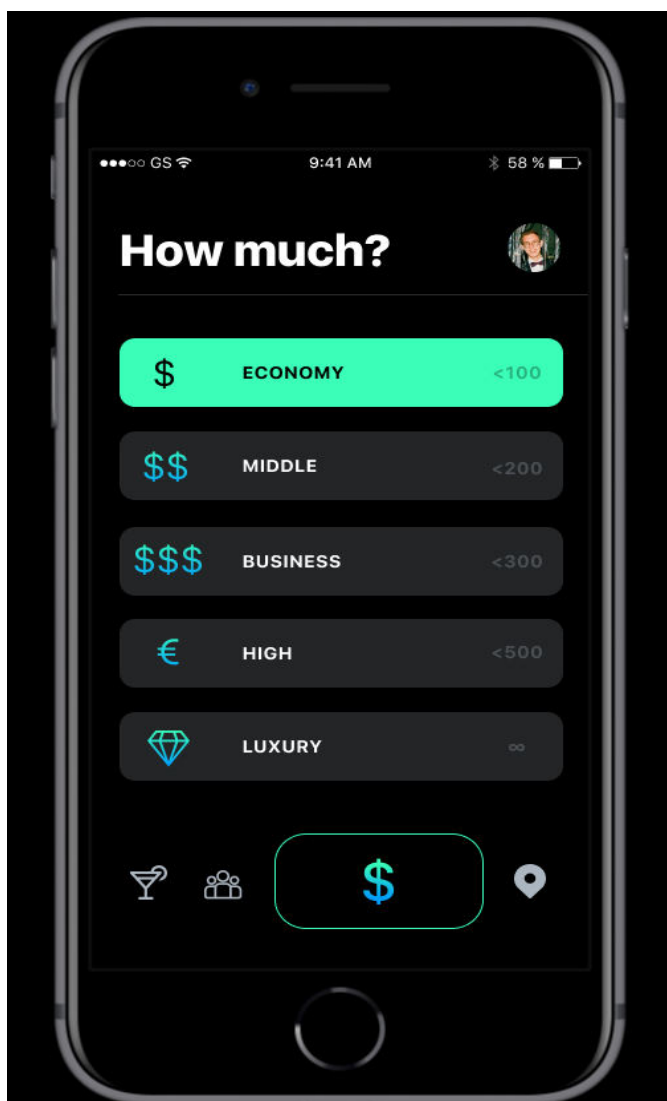


Рисунок 2.7 – Вибір бюджету на одну людину

2.4.4. Генерація закладу

Після обраних параметрів користувача зустрічає сторінка рис. 2.8 на якій користувач повинен вказати своє місце розташування та після натиску на кнопку «GO» просто вирушити до вказаного місця.

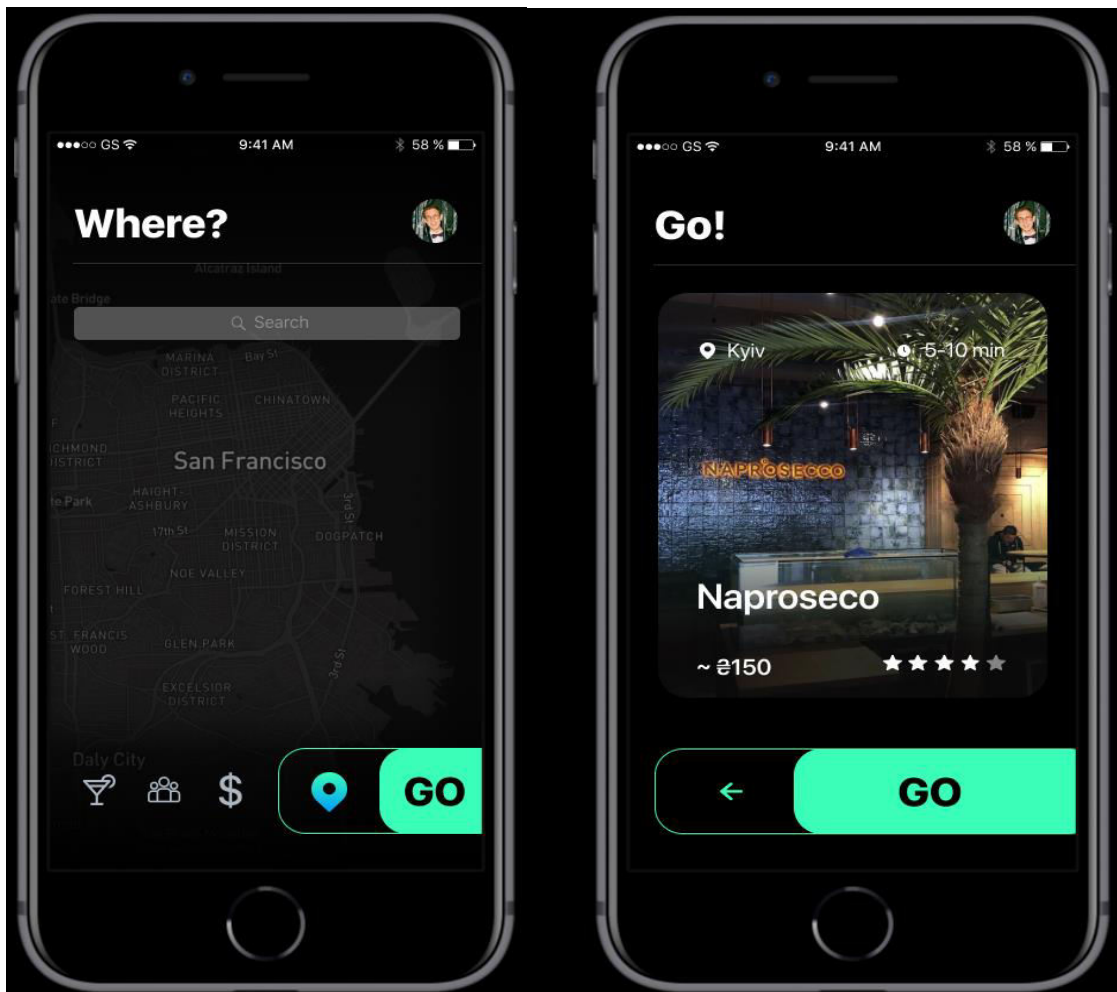


Рисунок 2.8 – Приклад генерації закладу

2.4.3. Пошук закладу на карті

Після того, як заклад був згенерований, користувач може знайти його на карті та прокласти маршрут до обраного місця рис.2.9.

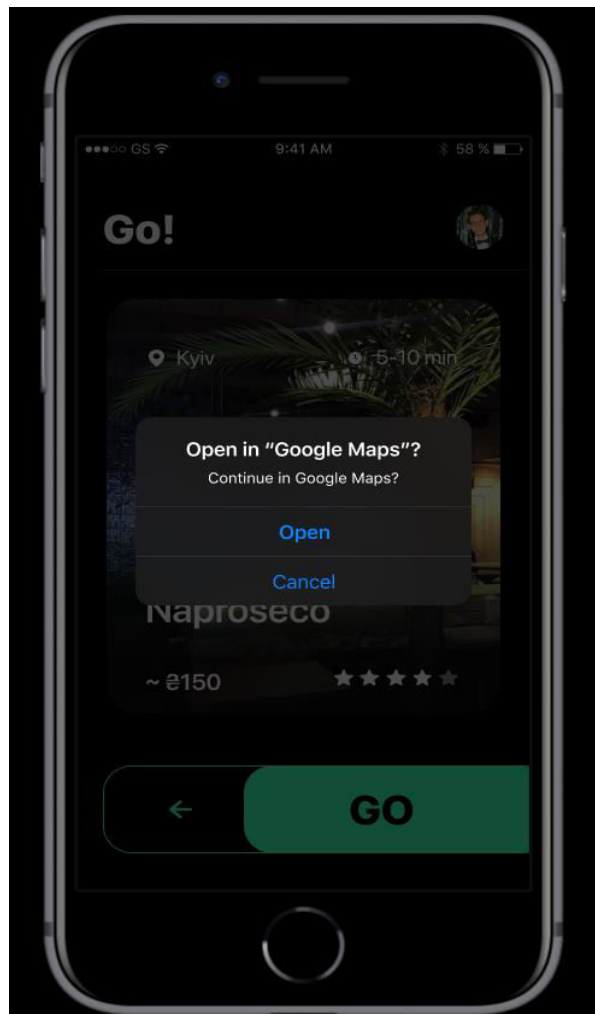


Рисунок 2.9 – Екран пошуку місця на карті

2.5. Прецеденти

Розглянемо виділені прецеденти, а саме :

- 1) Обрати сегмент закладу.
- 2) Обрати кількість людей.
- 3) Обрати середню вартість меню на одну людину.
- 4) Знайти згенерований заклад на карті.

2.5.1. Обрати сегмент закладу

Даний прецедент описує одну із головних параметрів додатку, він виконується на самому початку роботи користувача з програмою.

Прецедент має один сценарій використання:

- 1) Користувач обирає сегмент закладів з перелічених. Опис в табл.2.1

Таблиця 2.1. Опис прецеденту «Обрати сегмент закладу»

Дії користувача	Відгук системи
1. Користувач натискає кнопку обраного сегменту	2. Система розпочинає процедуру сортування всіх закладів в базі даних за обраним параметром
3. Користувач переходить до наступного прецеденту	4. Система чекає подальших дій користувача.

Альтернатива:

Якщо по якимось причинам користувач не обрав сегмент закладу та спробував перейти до наступного прецеденту, система покаже модальне вікно з попередженням для користувача, щоб той обрав сегмент закладу, модальне вікно буде викликатись доки користувач не вибере сегмент закладу .

2.5.2. Обрати кількість людей

Даний прецедент не є основним параметром пошуку закладу, тому користувач може пропустити його, але при цьому збільшується кількість закладів серед яких буде обраний підходящий. Сценарій може мати такі варіанти:

- 1) Користувач обрав кількість людей та обраний параметр для пошуку.
Опис даного варіанту в табл. 2.2
- 2) Користувач не обрав параметр пошуку закладу та перейшов до наступного прецеденту. Опис даного варіанту в табл. 2.3

Таблиця 2.2. Опис прецеденту «Обрати кількість людей»

Дії користувача	Відгук системи
1. Користувач натискає обрану кількість людей	2. Система розпочинає процедуру сортування всіх закладів в базі даних за обраним параметром (Якщо користувач обрав 1 або 2 параметр, система відбирає менш залюднені місця.)
3. Користувач переходить до наступного прецеденту	4. Система чекає подальших дій користувача.

Таблиця 2.3. Опис прецеденту «Обрати кількість людей»

Дії користувача	Відгук системи
1. Користувач не натискає обрану кількість людей	3. Система ігнорує пошук по заданому параметру
3. Користувач переходить до наступного прецеденту	4. Система чекає подальших дій користувача.

2.5.3. Обрати середню вартість меню на одну людину

Даний прецедент також не є основним параметром пошуку закладу, тому користувач може пропустити його, але при цьому збільшується кількість закладів серед яких буде обраний підходящий. Сценарій може мати такі варіанти:

- 1) Користувач обрав параметр для пошуку з представлених. Опис даного варіанту в табл. 2.4
- 2) Користувач не обрав параметр пошуку закладу та перейшов до наступного прецеденту. Опис даного варіанту в табл. 2.5

Таблиця 2.4. Опис прецеденту «Обрати вартість меню»

Дії користувача	Відгук системи
1. Користувач натискає обрану середню ціну.	4. Система розпочинає процедуру сортування всіх закладів в базі даних за обраним параметром .
3. Користувач переходить до наступного прецеденту	4. Система чекає подальших дій користувача.

Таблиця 2.5. Опис прецеденту «Обрати середню вартість»

Дії користувача	Відгук системи
1. Користувач не натискає обрану кількість людей	5. Система ігнорує пошук по заданому параметру
3. Користувач переходить до наступного прецеденту	4. Система чекає подальших дій користувача.

2.5.4. Знайти згенерований заклад на карті

Даний прецедент є фіналом сценарію користування додатком, він має такі шляхи:

- 1) Після того як користувач обрав всі параметри пошуку йому потрібно вказати де він знаходиться або ввімкнути геолокацію та натиснути кнопку «GO», система генерує заклад за обраними параметрами та пропонує відкрити його на карті. Опис в табл. 2.6.
- 2) Після того як користувач ввімкнув геолокацію або вказав місце де він знаходиться після натиску кнопки «GO» він може скасувати пошук закладу на карту для корегування параметрів пошуку або для генерації іншого закладу. Опис в табл. 2.7.

Таблиця 2.6. Опис прецеденту «Обрати вартість меню»

Дії користувача	Відгук системи
1. Користувач вказує місце де він знаходиться або вмикає геолокацію.	6. Система розпочинає процедуру генерації закладу за обраним параметром серед отсортованої бази закладів.
3. Користувач натискає кнопку «GO»	4. Система пропонує знайти заклад на карті.

Таблиця 2.7. Опис прецеденту «Обрати середню вартість»

Дії користувача	Відгук системи
1. Користувач вказує місце де він знаходиться	7. Система розпочинає процедуру генерації закладу за обраним параметром серед отсортованої бази закладів.
3. Після кнопки «GO»	4. Система повертає на минулий прецедент

користувач відмінив пошук на карті	
---------------------------------------	--

2.6. Архітектура серверної частини

Доцільно буде використання фреймворку Express для побудови архітектури серверної частини. За допомогою даного фреймворку інкапсулюється обробка об'єктів високого рівня додатку, такі як, об'єкти запиту та об'єкти відповіді, надаючи їх під контроль розробнику .

Структура серверної частини повторює принципи такого типового архітектурного рішення як - MVC. Розглянемо основні представлення компонентів серверної частини, які були розроблені.

Маршрутизатор (Route) – відправляє на подальшу обробку даних керуючим об'єктам нижчого рівня (Контролеру чи API).

Контролер (Controller) - цей компонент відповідальний за зв'язок між model і view. Код компонента controller дає інформацію про реакції сайту на дії користувача. По суті, це мозок MVC-додатки.

Модель (Model)- цей компонент відповідає за дані, а також визначає структуру програми. Надає інтерфейс для роботи з сутностями.

Вигляд (View) - представлення, цей компонент відповідає за взаємодію з користувачем. Тобто код компонента визначає зовнішній вигляд програми і способи його використання.

API - інтерфейс для роботи з даними предметної області. Дані повертаються у форматі JSON. В рядку URL визначається запит до API

параметром "api". Відповідно до REST, була побудована семантика запитів таким чином. таблиця 2.8.

Таблиця 2.8. Семантика запитів відповідно до REST

Метод	GET	POST	PUT	DELETE
/class	Повернути список усіх закладів	Залишити коментар	-	-
/class/:id	Повернути один заклад	-	Редагувати коментар	Видалити коментар

Отримання HTML документу завжди являється першим запитом клієнта до сервера. Після чого завантажується JavaScript і виконується на клієнті, при необхідності запитуючи дані у API серверної частини.

Буде доцільно завантажити всі документи при першому запиті. Це дозволить не виконувати запит на сервер для пошуку документів, пропускаючи першу точку доступу і отримати дані з локального сховища користувача. Проте це стане причиною проблеми довгого очікування завантаження даних. Тому було прийнято рішення не використовувати такий підхід.

Було вирішено використання проксі-сервера для надання статичних файлів клієнту – це зменшить час запиту. Nginx виступає в ролі проксі-сервера і чудово справляється з цією роботою.

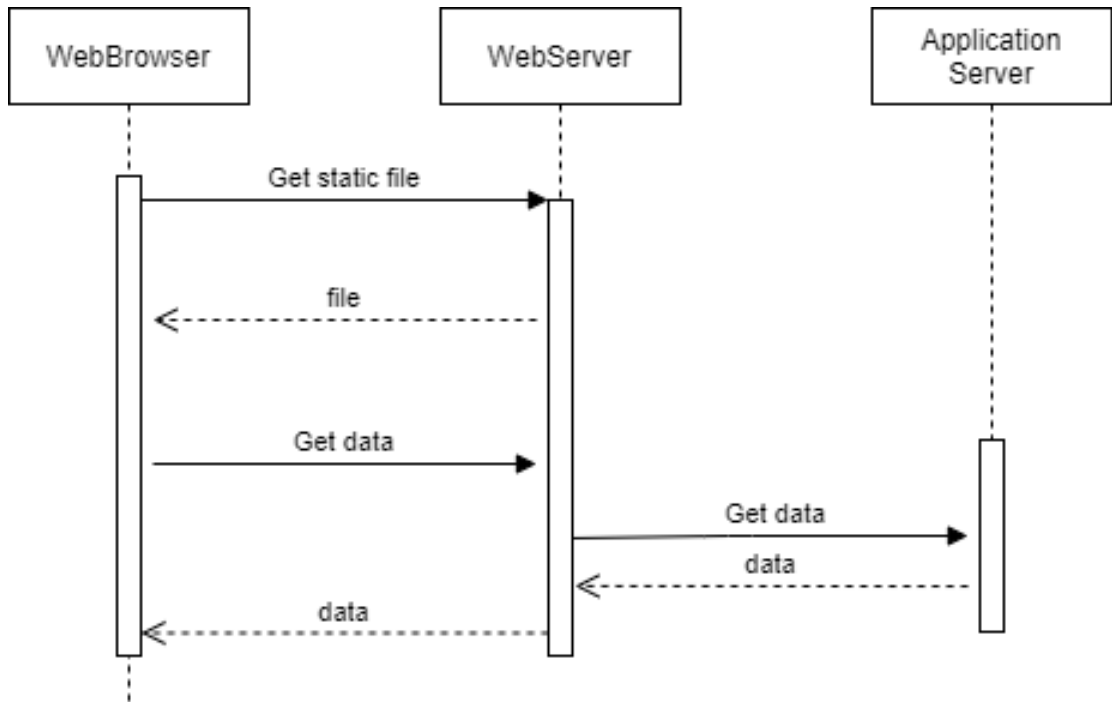


Рисунок 2.10 – Обробка запитів проксі-сервером

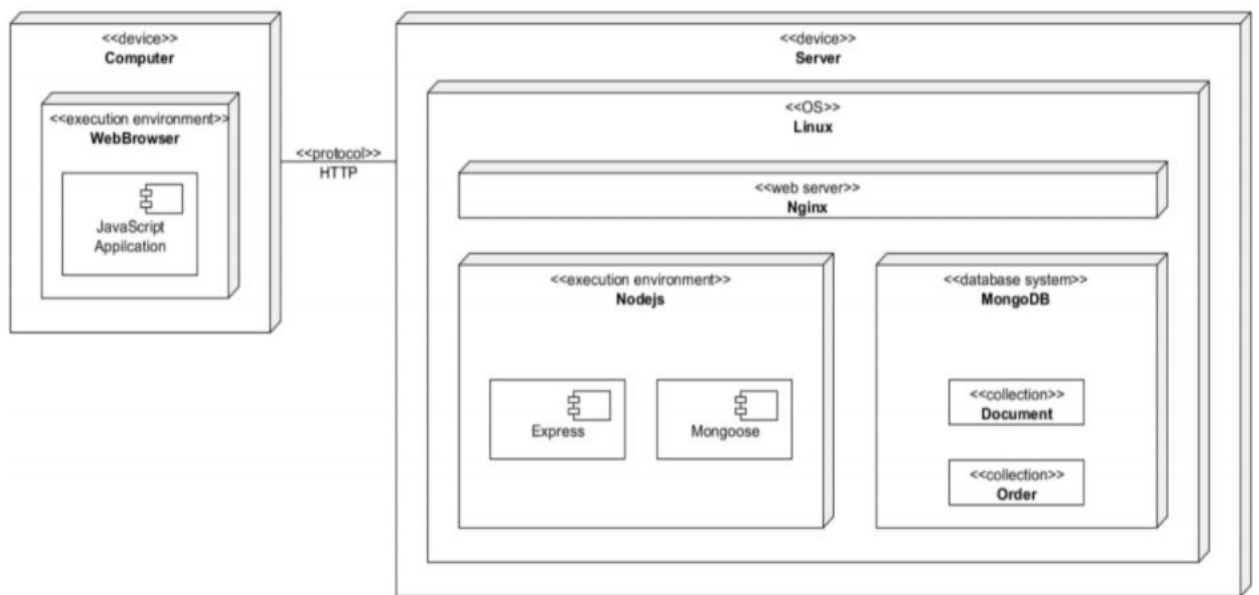


Рисунок 2.11 – Діаграма архітектури додатку

ВИСНОВОК ДО РОЗДІЛУ 2

Даний розділ магістерського проекту було присвячено опису предметної області завдання, проведенню формування основних вимог та базового інструментарію для розроблювального застосунку. Було опрацьовано сценарій використання додатку користувачем та створений первинний макет дизайну інтерфейсу юзера.

Після проведеного аналізу та передпроектування додатку були виділені основні прецеденти, якими є:

- 1) Обрати сегмент закладу.
- 2) Обрати кількість людей.
- 3) Обрати середню вартість меню на одну людину.
- 4) Знайти згенерований заклад на карті.

Ці прецеденти були опрацьовані більш конкретно, так як вони у змозі мати різні шляхи розвитку в залежності від вибору користувача. Врахування таких випадків на етапі проектування зробить подальшу розробку більш простою та надасть змогу запобігти виникненню небажаних помилок під час роботи додатку. Також в даному розділі була описана архітектура серверної частини та наведений приклад взаємодії серверу та клієнта.

РОЗДІЛ 3

РОЗРОБКА ДОДАТКУ

3.1. Обґрунтування вибору технологій

3.1.1. Вибір платформи для додатку

Одним із основних пріоритетів розробки додатку є досягнення максимальної доступності для користувачів з різними типами пристроїв та операційних систем на них. Для виконання цих завдань були розглянуті дві найпопулярніші мобільні платформи (IOS та Android).

IOS (iPhone OS) - це ОС для пристроїв Apple, розроблена компанією Apple Inc. На даній операційній системі працюють мобільні пристрої, які були випущені за час існування компанії. IOS займає друге місце після Android по кількості пристроїв які були продані на базі ОС.

Станом на січень 2020 року в App Store розміщено понад 3 мільйонів програм для використання під даною операційною системою, 1 000 000 з яких були розроблені для підтримки iPad. Ці мобільні додатки завантажено понад 200 мільярдів разів. Чверть світового ринку мобільних пристроїв працюють на базі IOS.

Android - це ОС, розроблена Google, в основі Android лежить ОС Linux і орієнтована в основному на мобільні пристрої. Реалізація Dalvik - Java Virtual Machine є основним компонентом цієї ОС, розроблене програмне забезпечення засновані на даній реалізації Java. За інформацією, наданою Google за 2019 рік, компанія має більше 1.5 мільярдів активних користувачів на місяць. У квітні 2017 року операційна система Android стала найпопулярнішою ОС у всьому світі. За офіційними даними, ОС Android

займає понад 70% ринку мобільних пристроїв. Для охоплення максимальної кількості цільових користувачів буде доцільно розробити кроссплатформенний додаток, який би працював як на Android так і на IOS. Реалізація даного вибору буду описано нижче у розділі.

3.1.2. Вибір мови програмування

Для реалізації кроссплатформеного мобільного додатку доцільно обрати мову програмування Java Script. Ця мова є лідером в розробці веб та мобільних додатків на сьогоднішній день. Також вибір пав на цю мову програмування тому що, на сьогоднішній день Java Script має підтримку користувачів та велику кількість Open Source бібліотек і фреймворків , які прискорюють, оптимізують, та полегшають написання подібних додатків.

3.1.3. Вибір допоміжних бібліотек та фреймворків

Отже для реалізації даного додатку було обрано наступні бібліотеки:

- 1) React
- 2) React Native
- 3) Redux
- 4) Passport.js
- 5) Node.js
- 6) Express

React — opensource бібліотека для Java Script, розроблена для полегшення створення користувацьких інтерфейсів (UI), та для вирішення проблеми частиного оновлення веб-сторінки, з якими зустрічаються під час

розробки односторінкових веб-додатків. Розроблювався та підтримується компанію Facebook.

За допомогою React розробникам стає простіше розробка великих веб-додатків, які використовують динамічні дані, які змінюються з певним плином часу, без перезавантаження самої веб-сторінки. Основна ідея бібліотеки в тому що, вона легка, масштабована та швидка. React обробляє тільки інтерфейс користувача у додатках. Це відповідає *вигляду(view)* у шаблоні MVC архітектури, а також використовується разом з іншими бібліотеками JavaScript або в великих фреймворках MVC, наприклад AngularJS. Він також використовується з React для надбудов. Як бібліотеку UI React частіше використовують разом з іншими бібліотеками, наприклад Redux. Аргументи передаються до функції рендерер, як аргументи HTML тегу. Компонент не повинен напряду змінювати аргументи, які були йому передані, але є змога їх змінювати через callback функцію. Такий підхід називають «властивості вниз, події вверх».

React використовує віртуальний DOM, а не звичайний DOM браузера. Це дозволяє знаходити частини DOM які змінилися на стороні клієнта, порівняти з останньою версією віртуального DOM, і визначити, як краще всього змінити DOM браузера. Таким чином розробник працює з веб-сторінкою, думаючи що вона оновлюється повністю, але бібліотека вирішує самотійно які компоненти сторінки треба оновити а які ні. Компоненти React майже завжди написані на JSX.

Методи життєвого циклу – це методи, які вбудовані за допомогою React. Вони допомагають розробнику змінювати та редагувати дані в різних позиціях життєвого циклу додатку React. Наприклад:

- `shouldComponentUpdate` – це метод життєвого циклу, який каже Javascript, що потрібно оновити компонент, за допомогою логічних змін.
- `componentWillMount` – це метод життєвого циклу, який каже Javascript, що потрібно налаштувати деякі дані перед вставленням компонентів у віртуальний DOM.
- `componentDidMount` – це метод життєвого циклу, схожий на компонент `WillMount`, єдина різниця, що він працює після методу `render`, і може використовуватися для додавання JSON-об'єктів, а також для визначення атрибутів та станів.
- `render` – це найважливіший метод життєвого циклу, необхідний у будь-якому компоненті. Метод `render` з'єднується з JSX і відображає власний JSX на веб-сторінці.

Приклад JSX:

```
class App extends React.Component {  
  render() {  
    const i = 1;  
    return (  
      <div>  
        <h1>{ i === 1 ? 'true' : 'false' }</h1>  
      </div>  
    );  
  }  
}
```


Код, який був написаний на JSX, перетворюється за допомогою інструменту, Babel. Це перетворення, виконується під час збірки проекту, перед запуском програми.

React Native представлений компанією Facebook у 2015 році, він використовує React-архітектуру для створення нативних IOS, Android та UWP додатків. Принципи роботи React Native повністю повторюють React, єдина різниця це те що, він не використовує DOM через VirtualDom. Він запускається у фоновому режимі прямо на пристрої і контактує з нативною платформою.

React спрямований на створення інтерфейсів користувача, і тому не має в своєму складі багато інструментів, які деякі розробники вважають лишніми. Це дозволяє обрати використання бібліотек які на розгляд розробника потрібно використовувати, для виконання певних поставлених задач.

Redux - бібліотека для JavaScript з відкритим кодом, призначена для контролю стану додатку. Найчастіше використовується з React або Angular для розробки клієнтської частини. Містить ряд інструментів, що дозволяють значно спростити передачу даних сховища через контекст.

Passport - це програмне забезпечення для реалізації автентифікації для Node.js. Дуже гнучкий та модульний, Passport можна ненав'язливо запустити в будь-якій веб-програмі на базі Express. Комплексний набір стратегій реалізує автентифікацію за допомогою імені користувача та пароля, Facebook, Google та інші. Він створений для однієї мети: автентифікації запитів. Під час написання модулів інкапсуляція є перевагою, тому Passport делегує всі інші функції додатку. Таке розділення забезпечує чіткість та коду та робить Passport надзвичайно простим для інтеграції в додаток.

У сучасних веб-додатках автентифікація може приймати різні форми. Традиційно користувачі входять, вводячи ім'я користувача та пароль. З розвитком соціальних мереж єдиний вхід за допомогою постачальника OAuth, такого як Facebook або Twitter, став популярним методом автентифікації. Служби, які надають API, часто вимагають облікових даних на основі маркерів для захисту доступу.

Passport визнає, що кожна програма має унікальні вимоги щодо автентифікації. Механізми автентифікації, відомі як стратегії, упаковані як окремі модулі. Додатки можуть вибирати, які стратегії використовувати, не створюючи зайвих залежностей.

Незважаючи на складність автентифікації, код не повинен бути складним.

Аутентифікація запитів проста, як виклик `passport.authenticate()` та вказівка, яку стратегію використовувати. Підпис функції `authenticate()` є стандартним підключенням проміжного програмного забезпечення, що робить його зручним для використання як проміжне програмне забезпечення маршруту в додатках Express. За замовчуванням, якщо автентифікація не вдається, Passport відповість статусом 401 Несанкціонований, і будь-які додаткові обробники маршруту не будуть викликані. Якщо автентифікація буде успішною, буде викликано наступний обробник, а властивість `req.user` буде встановлено для аутентифікованого користувача.

Примітка: Стратегії потрібно налаштувати перед використанням їх у маршруті. Перенаправлення зазвичай видається після автентифікації запиту. У цьому випадку параметри переспрямування замінюють поведінку за замовчуванням. Після успішної автентифікації користувач буде

перенаправлений на домашню сторінку. Якщо автентифікація не вдається, користувач буде перенаправлений назад на сторінку входу для нової спроби.

Для використання паспорта для автентифікації потрібно налаштувати три частини:

- 1) Стратегії аутентифікації
- 2) Проміжне програмне забезпечення програми
- 3) Сеанси (необов'язково)

Passport використовує так звані стратегії для автентифікації запитів. Стратегії варіюються від перевірки імені користувача та пароля, делегованої автентифікації за допомогою OAuth або федеративної автентифікації за допомогою OpenID.

Для запуску коду JS на сервері, його потрібно інтерпретувати в машинний код. Тому була обрана платформа NodeJS, як середовища виконання JS коду, тому що:

- Працює згідно асинхронної моделі.
- Добре задокументований.

NodeJS - платформа для розробки додатків, які націлені на високу продуктивність вводу-виводу і малу інтенсивність обчислень. За допомогою платформи можлива розробка серверних додатків на JS і можливість взаємодіяти з пристроями введення-виведення завдяки API NodeJS .

NodeJs використовує транслятор розроблений компанією Google, який дозволяє перетворити код JavaScript в асемблер цільового процесора, що забезпечує високу продуктивність .

Nodejs має доступ до ресурсів сервера, які є необхідними для реалізації додатку. Модулі, розроблені на NodeJS здатні створювати нові процеси . Наприклад, Nodejs має модуль веб-сервера для обробки HTTP-запитів.

Для створення гнучкої настройки, важливо вибрати відповідний фреймворк . Тому був обраний Express, так як він має такі переваги:

- Дуже висока швидкість обробки запитів, яка підвищить швидкість роботи сервера.
- Популярність і документація.

3.2 Рішення обміну даними

3.2.1 API

У якості методу організації рішення проблеми обміну даними з API, було обрано шаблон проектування REST, тому що:

- Він має просту семантику.
- Є універсальним інтерфейсом.
- Дозволяє використовувати HTTP-клієнти.

При використанні REST, URL визначає семантику запитів. REST є більш сучаснішою альтернативою складним способам обміну даними, таким як SOAP або RPC.

3.2.1 Форма обміну даними

Як формат для обміну даних між клієнтом і сервером, було обрано формат JSON, тому що:

- Він підтримується усіма популярними браузерами.
- В нативному форматі JavaScript

3.2.3. Транспортування даних

HTTP був обраний як надійний протокол обміну даними, він є загальноприйнятим і доступним для запитів до API. Об'єкт XMLHttpRequest був обраний для використання AJAX технології, яка має підтримку більшої кількості бібліотек. Якщо говорити точніше буде використаний сучасний метод , який називається Fetch API.

3.3. База даних

Дані додатку необхідно постійно обробляти та зберігати, для цієї задачі використовують базу даних. Перш ніж обрати СУБД, слід розглянути передумови вибору.

Так як місця та заклади можуть відрізнятися атрибутами, в залежності від типу, було прийнято рішення обрати СУБД, де є підтримка неструктурованою моделі даних - NoSQL .

Базуючись на передумовах, було прийнято рішення в сторону СУБД, яка має:

- Підтримку NoSQL.

Великим плюсом є підтримка JSON-формату, тому що даний формат обраний для обміну даними між клієнтом і сервером..

Обраною СУБД була MongoDB на підставі вхідних запитів.

MongoDB - документно-орієнтована СУБД . Зберігає дані в BSON-форматі, що представляє собою бінарну форму подання JSON. Таким чином, запис в MongoDB є аналогом об'єктного представлення у форматі JSON.

MongoDB має інтерфейс для запуску операцій на JavaScript, підтримує потрібні типи даних, які можуть знадобитися при проектуванні БД, наприклад, масив, дата, об'єкт чи інші.

3.4. Сервер

Як мову програмування для розробки серверної частини був обраний JavaScript за наступними причинами:

- Розповсюдженість і документація.
- MongoDB має інтерфейс запитів на JavaScript.
- JSON нативний формат JavaScript і використовується для обміну даними між клієнтською частиною та API серверної частини.
- Використання однієї мови JavaScript на сервері і при роботі з СУБД,.

JavaScript - прототипно-орієнтована мова програмування. Це означає , що вона є обмеженою об'єктно-орієнтованою мовою (немає поняття класу і успадкування). Саме тому, підхід до об'єктно-орієнтованого програмування імітуються, наслідування - це копія батіківського об'єкта з додаванням до нього нової поведінки або властивостей. Відношення будується за принципом "прототип - наслідник" збереження структури прототипу не є обов'язковим. Завдяки цьому, JavaScript дозволяє імітувати об'єктно-орієнтований підхід до розробки.

3.4.1 Веб-сервер

У якості веб-сервера буде використовуватися Nginx для передачі клієнту статичних файлів, а для отримання динамічних даних буде передаватися запит серверу на NodeJS. Nginx буде виступати як проксі-сервер, для прискорення роботи додатку.

Nginx - веб-сервер і поштовий проксі-сервер, що працює на Unix-схожих ОС.

Nginx був обраний тому що:

- Використовується як проксі-сервер.
- Гнучкість налаштування.
- Поширеність.
- Доступність.

3.4.2 Віртуальний виділений сервер

VDS було вирішено використовувати на публічній хмарі, через зручність використання.

Публічна хмара - інфраструктура, призначена для вільного використання широкою публікою. Хмара може перебувати у власності, керуванні та використанні комерційних, наукових і урядових організацій.

Існує три головних конкурента які виділяються у представлених технологіях: Google Cloud Platform, Amazon Web Services, Microsoft Azure.

Таблиця 3.1. Порівняння постачальників хмарних технологій

Назва	Переваги	Недоліки
AWS	<ul style="list-style-type: none"> • Фаворит на ринці послуг • Зрілі пропозиції • Підтримка великих організацій • Широка документація • Глобальне охоплення 	<ul style="list-style-type: none"> • Тяжке в використанні • Управління витратами
Microsoft Azure	<ul style="list-style-type: none"> • Другий найбільший постачальник • Інтеграція з інструментами 	<ul style="list-style-type: none"> • Проблеми з документацією

	та програмним забезпеченням Microsoft <ul style="list-style-type: none"> • Широкий набір функцій • Гібридне хмара • Підтримка відкритого коду 	<ul style="list-style-type: none"> • Неповні інструменти управління
Google	<ul style="list-style-type: none"> • Розроблений для підприємств, які працюють у хмарі • Прихильність до відкритих джерел та портативності • Експертиза DevOps 	<ul style="list-style-type: none"> • Пізній учасник ринку IaaS • Менше можливостей та послуг

Було обрано Amazon Web Services в якості постачальника послуг, тому що він має:

- Детальну та зручну документацію
- Підтримку
- Різноманітність хмарних послуг

ВИСНОВОК ДО РОЗДІЛУ 3

Цей розділ магістерської дисертації був присвячений перегляду та аналізу аспектів вибору стека технологій та бібліотек для розробки прогнозованого додатку, обґрунтування вибору цільової платформи для розробки додатків, мови написання програми та пристроїв, які будуть підтримуватися за додатком. Враховуючи вимоги вимови, було обрано мову програмування та платформу та допоміжні бібліотеки, що значно пришвидшило б розвиток та дозволило б реалізувати універсальну архітектуру з максимально високим рівнем оптимізації. Було розглянуто кожну технологію та обґрунтоване пояснення вибору тої чи іншої технології та бібліотеки.

РОЗДІЛ 4

РОЗРОБКА СТАРТАП-ПРОЕКТУ

4.1. Опис ідеї проекту

В табл. 4.1 приводиться зміст головної ідеї проекту, потенційні шляхи практичного застосування продукту та базові привілегії, які користувач зможе отримати при використанні розроблюваного додатку.

Таблиця 4.1. Опис ідеї стартап-проекту

<i>Зміст ідеї</i>	<i>Напрямки застосування</i>	<i>Вигоди для користувача</i>
Розробка зручного та швидкого додатку.	Легкий та швидкий туризм	Заощадження часу на вибір закладу

Далі буде проводитися аналіз технічно-економічного потенціалу та параметрів розробки та подальше порівняння з конкурентними додатку. Основними конкурентами є TripAdvisor, Yelp, 4sq.

Таблиця. 4.2.

Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№	Техніко-економічні характеристики проекту	Мій проект	Конкуренти			W	N	S
			1	2	3			
1	Швидкий пошук закладу	+	-	-	-	+		
2	Легкість в використанні	+	+	-	+		+	
3	Кросплатформеність	+	+	+	+		+	
4	Елементи соціальної мережі	-	+	+	+			+

4.2. Технологічний аудит ідеї проекту

Тепер проведемо аудит технологічних засобів, за використанням яких буде реалізовано проект. Результати наведено в табл. 4.3.

Таблиця 4.3. Технологічна здійсненність ідеї проекту

№	Ідея проекту	Технології для реалізації	Наявність технологій	Доступність технологій
1	Додаток для швидкого пошуку закладу	React, ReactNative	У наявності	У відкритому доступі
2		Node.js	У наявності	У відкритому доступі
3		Mongo DB	У наявності	У відкритому доступі
4		JavaScript	У наявності	У відкритому доступі

5		Passport	У наявності	У відкритому доступі
6		Redux	У наявності	У відкритому доступі
7		Express	У наявності	У відкритому доступі

4.3. Аналіз ринкового потенціалу для старту стартап-проекту

Наступним кроком стане аналіз ринку, що наведений у табл. 4.4.

Таблиця 4.4. Попередня характеристика потенційного ринку стартап-проекту

№	Показники стану ринку	Характеристика
1	Кількість головних гравців, од	5-7
2	Загальний обсяг продажу, грн/ум.од	Середня ціна за ліцензію на використання складає 10-20 тис. доларів
3	Динаміка ринку	Постійний ріст
4	Наявність обмежень для входу	Середній рівень конкуренції, при виборі рішення з безпеки довіряють в основному компаніям з репутацією
5	Специфічні вимоги до стандартизації та сертифікації	Не передбачено чинним законодавством

У таблиці 4.5 приводяться можливі користувацькі групи, їх вимоги та потреби від додатку.

Табл. 4.5 Характеристика можливих користувачів стартапу

<i>№</i>	<i>Потреба формує ринок</i>	<i>Цільова аудиторія проекту</i>	<i>Різниця поведінки цільових груп</i>	<i>Вимоги користувачів до розробки</i>
1	Потреба в заощадженні часу на виборі закладу для проведення часу	Користувачі смартфонів	Необхідність задоволення в використанні додатку	Зручність використання та швидкість роботи

Після визначення можливих цільових груп користувачів ринок було проаналізовано на наявність факторів загроз (табл. 4.6) та можливостей (табл. 4.7).

Таблиця 4.6. Фактори загроз

<i>№</i>	<i>Фактор</i>	<i>Зміст загрози</i>	<i>Можлива реакція компанії</i>
1	Можлива конкуренція	Наявність на ринку додатків з використовуваними технологіями	Розширення функціоналу стартап-проекту
2	Не велика потреба у програмі	Можливість важкого старту запуску додатку у масове використання	Проведення маркетингової компанії та демонстрація корисності додатку

Таблиця 4.7 Фактори можливостей

<i>№</i>	<i>Фактор</i>	<i>Зміст можливості</i>	<i>Можлива реакція компанії</i>
1	Залучення інвестицій	Можливість залучення інвестицій від інвестиційних фондів та інших компаній	Можливість розширення штату розробників, розширення функціоналу додатку

Загальні фактори конкуренції приводяться у таблиці 4.8.

Таблиця 4.8. Ступеневий аналіз конкуренції ринку

<i>Специфіка конкурентного становища</i>	<i>Проява характеристики</i>	<i>Вплив на вектор дій компанії (потенційні кроки для підтримки конкурентоспроможності)</i>
1. Тип конкуренції: чиста	Наявність незалежної конкуренції на ринку	Націлення на технології та стрімкий розвиток
2. Рівень конкуренції: міжнаціональна	Конкуренція з міжнародними компаніями	Робота на міжнародному ринку
3. За галузевою ознакою: внутрішньогалузева	Випуск широконапрявленого програмного забезпечення	Фокусування на ринку електронних гідів
4. Конкуренція товарним видом: товарно-видова	Наявність відомих конкурентів у сфері	Отримання авторитету влаштовуючи спеціалістів сфери поточного проекту
5. За інтенсивністю: марочна	Наявність відомих конкурентних брендів	Створення власного відомого та визнаваного бренду пробукту

Виконаємо аналіз конкурентних умов по моделі М. Портера (табл. 4.9).

Таблиця 4.9. Аналіз конкуренції в галузі за М. Портером

<i>Складові аналізу</i>	<i>Прямі конкуренти на ринку</i>	<i>Можливі конкуренти на ринку</i>	<i>Постачальники</i>	<i>Клієнти</i>	<i>Товари-замінники</i>
	TripAdvisor, 4sq	TripAdvisor, 4sq	TripAdvisor, 4sq	Середні та малі компанії	Немає
<i>Висновки</i>	Достатня інтенсивність конкурентної боротьби	Конкуренція на ринку має місце за умови підтримки конкурентноздатних умов	Постачальники диктують значну вартість ліцензій	Значні клієнтські вимоги до виробників	Відсутність товарів для заміни продукту

Функціонування на ринку має можливість, але для цього слід забезпечити конкурентну вартість та перелік можливостей розробленого продукту.

Узагальнення конкурентоспроможних аспектів показано у табл. 4.10.

Таблиця 4.10. Узагальнення аспектів конкурентоспроможності

<i>№</i>	<i>Аспект конкурентоспроможності</i>	<i>Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)</i>
1	Підтримка мобільних можливостей	Основним ринком для інтеграції є ринок мобільних платформ та додатків

2	Кросплатформеність	Розробка на мові програмування JS надає можливість використання його будь-яким пристроєм під ОС Android та IOS.
3	Ціна ліцензії	Встановлення рентабельної ціни у порівнянні із потенційними конкурентами

Наступним кроком проведемо зіставлення сильних та слабких сторін проекту порівняно із конкуруючими розробками, його наведено в табл. 4.11.

Таблиця 4.11. Зіставлення сильних та слабких сторін продукту

№	Аспект конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів						
			-3	-2	-1	0	1	2	3
1	Підтримка мобільних можливостей	19						x	
2	Кросплатформеність	18					x		
3	Ціна ліцензії	15			x				

Фінальним кроком є SWOT-аналіз (табл. 4.12).

Таблиця 4.12. SWOT-аналіз стартап-проекту

Сильні сторони: кросплатформеність, простота та доступність у використанні	Слабкі сторони: менший функціонал додатку у порівнянні із конкуруючими продуктами
Можливості: швидкий пошук закладу	Загрози: відчутна ринкова конкуренція

Базуючись на проведеному аналізі були сформовані альтернативні варіанти дій на ринку та заходи їхньої інтеграції (рис. 4.13).

Таблиця 4.13. Альтернативи ринкового впровадження стартап-проекту

<i>№</i>	<i>Альтернативний варіант ринкової стратегії</i>	<i>Шанс отримати ресурси</i>	<i>Термін впровадження</i>
1	Зацікавлення проектів інших компаній на ринку	Продаж розробки або її поглинання зацікавленими особами	Декілька років
2	Перетворення розробки на лад проекту із відкритим вихідним кодом	Потенціал залучення розробників та експертів з області інформаційних технологій	Менше одного року
3	Продаж реалізованих у проекті технологій та рішень	Надання можливості розвитку на платформах, не підтримуваних конкурентами нижчою вартістю	Кілька місяців

Зазначимо обрану альтернативу: продаж реалізованих у проекті технологій та рішень.

4.4. Розробка ринкової стратегії стартап-проекту

В табл. 4.14 наведено опис цільових груп потенційних споживачів продукту. Такими групами стали малий, середній та великий бізнес у різних галузях інформаційних технологій.

Таблиця 4.14. Вибір цільових груп потенційних споживачів

<i>№</i>	<i>Опис основної маси клієнтів</i>	<i>Готовність клієнтів до сприйняття продукту</i>	<i>Очікуваний попит основної групи</i>	<i>Рівень конкуренції у сфері ринку</i>	<i>Простота входу у сферу ринку</i>
1	Малий бізнес	Готові	Високий попит	Достатня	Проста
2	Середній бізнес	Готові	Високий попит при встановленні помірної ціни	Середня	Середня
3	Великий бізнес	Не готові	Високий, необхідний більш високий рівень складності та розвитку розроблюваного проекту та застосованих технологій	Висока	Складна

Проаналізувавши отримані результати було прийнято рішення для розвитку обрати групи малого та середнього бізнесу, так як у масштабі великого бізнесу рішення приймаються повільно та вимагають підписань багатьох паперів, що буде складно в процесі масштабування та підтримки розробки.

Для нормального функціонування в обраних ринкових сегментах слід обрати основну стратегію розвитку (табл. 4.15).

Проаналізувавши отримані результати було прийняте рішення, альтернативним шляхом буде продажу своєї реалізації та технологій за меншою рентабельною ціною.

Таблиця 4.15. Обрання основної ринкової стратегії

<i>№</i>	<i>Обрана альтернатива розвитку</i>	<i>Стратегія захоплення ринку</i>	<i>Ключові конкурентні позиції згідно з обраною альтернативою</i>	<i>Базова стратегія розвитку</i>
1	Продаж своїх реалізацій та технологій	Зниження ціни та подальша співпраця з платформами меншої конкуренції	Зниження собівартості розробки та фокус на потреби широкого спектру користувачів	Концетрований маркетинг

У таблиці 4.16 сформовано основну концепцію конкурентноспроможного розвитку розроблюваного проекту.

Таблиця 4.16. Формування основної концепції конкурентного розвитку

<i>№</i>	<i>Чи виступає проект «першопрохідцем» на ринку?</i>	<i>Чи буде компанія знаходити нових користувачів, або відбирати у конкурентів їх аудиторію?</i>	<i>Чи буде компанія використовувати напрацювання конкуруючих розробок?</i>	<i>Стратегія конкурентної поведінки</i>
1	Так	Пошук нових користувачів серед цільової аудиторії	Ні, лише власні розробки	Заняття конкурентної ніші

Базуючись на основних вимогах користувачів до розроблюваного продукту, обраній стратегії конкурентної поведінки на ринку та основній концепції конкурентного розвитку розробки визначимо стратегію позиціонування (табл. 4.17).

Таблиця 4.17. Визначення стратегії позиціонування

<i>№</i>	<i>Потреби цільової аудиторії</i>	<i>Основна стратегія розвитку</i>	<i>Основні конкурентоздатні аспекти розроблюваного проекту</i>	<i>Параметри, що формують комплексну позицію розробки</i>
1	Простота та легкість використання додатку	Залучення до розробки прогресивних спеціалістів в області ІТ	Невелика ціна ліцензії на використання	Доступність, простота у використанні та низька ціна

4.5. Розроблення маркетингової програми стартап-проекту

На старті слід розробити маркетингову концепцію розробки, отримуваної користувачем. Отримані результати аналізу наведені в таблиці 4.18.

Таблиця 4.18. Узагальнення переваг концепції розроблюваного продукту

№	Запит	Корисність товару для користувача	Основні переваги перед конкурентами
1	Розвага	Розважальна складова продукту	Новизна напрямку використання технологій

Створимо трьохрівневу маркетингову модель товару (таблиця 4.19.)

Таблиця 4.19. Опис трьох рівнів моделі товару

Рівні	Сутності та складові	
1. Задум товару	Розробка зручного та розважального додатку	
2. Реальне виконання товару	Характеристики	Розмір
	Сайт, мобільний додаток	
	Якість: тестування з використанням юніт та інтеграційних тестів вздовж розробки продукту	
	Пакування: відсутнє	
	Марка: назва організації-розробника + назва товару	
3. Підкріплення товару	До продажу: код продукту	
	Після продажу: права на володіння	
Розроблюваний товар буде захищено від копіювання шляхом завчасної компіляції: оскільки розробка уже скомпільована, отримати її вихідний код стає важкою задачею. Також наявна користувацька ліцензійна домовленість та прямі договори, що забороняють копіювання розробки та плагіат		

У таблиці 4.20 було встановлено цінові рамки, що стануть ключовими аспектами при ціноутворенні продукту.

Таблиця 4.20. Визначення цінових рамок

<i>№</i>	<i>Діапазон цін товарів-замінників</i>	<i>Діапазон цін товарів- аналогів</i>	<i>Рівень прибутку основної групи споживачів</i>	<i>Верхня та нижня межа ціноутворення на товар/послугу</i>
1	\$15000-30000	\$5000-40000	>\$400000	\$0-\$8000

Систематику збуту наведено у таблиці 4.21. Було прийнято рішення проводити збут власними силами.

Таблиця 4.21. Формування системи збуту

<i>№</i>	<i>Особливості ринкової поведінки цільових клієнтів</i>	<i>Функції збуту постачальника товару</i>	<i>Глибина каналу збуту</i>	<i>Оптимальна система збуту</i>
1	Отримання ліцензії на програмну розробку шляхом купівлі	Підписання контрактів на збут	Однорівневий	Вертикальна, згідно з якою право власності на розробку залишається у розробника поточного програмного продукту

Останнім етапом є визначення концепції маркетингових комунікацій на ринку, що наведено у таблиці 4.22.

Таблиця 4.22. Концепція маркетингових комунікацій на ринку

<i>№</i>	<i>Особливості поведінки цільових клієнтів</i>	<i>Канали зв'язку, що будуть використовуватись цільовою клієнтурою</i>	<i>Ключові позиції, обрані для позиціонування</i>	<i>Основна ціль рекламної кампанії</i>	<i>Концепція рекламної кампанії</i>
1	Єдиноразове придбання шляхом підписання контракту	Сарафанне радіо, розміщення реклами	Зручний та швидкий додаток	Демонстрація програмних прикладних можливостей продукту	Виділення продукту серед конкурентів

ВИСНОВОК ДО РОЗДІЛУ 4

У даному розділі описується та виконується документування та аналіз стартап-проекту, визначаються та демонструються потенційно слабкі та сильні сторони ідеї, визначено потенційну технічну можливість його розробки.

Було з'ясовано, що ринок володіє високим рівнем конкуренції з боку великої кількості компаній, однак ціна ліцензії на застосування їх розробок також є достатньо високою.

Був проведений аналіз потенційної стратегії ринку орієнтуючись на малий та середній бізнес, відкидаючи варіант націлення на великий бізнес через вже існуючі їх взаємодії із бувалими гравцями на ринку.

Згідно із стратегією позиціонування було зроблено висновок, що існує необхідність сфокусуватись на розробці зручного та популярного додатку, та регулярному розширенні можливостей функціоналу стартап-проекту, а також існує необхідність установлення рентабельної низької ціни на продукт для успішного та швидкого входу на ринок та підтримання здорової чесної конкуренції.

ВИСНОВКИ

Дана магістерська дисертація присвячена дослідженню способів створення веб-додатку «Click&GO». Було представлено Мета та Ціль дослідження. А саме розробка та дослідження розширення функціоналу додатку. Створення зручного та юзерфрендлі додатку для швидкого пошуку місця та закладу для проведення свого часу.

Було виконане дослідження предметної області, визначено вимоги та задачі, з якими повинен справлятися додаток, наведено перелік прецедентів та шляхів використання застосунку. Основні прецеденти були розглянуті більш детально разом із потенційно можливими альтернативними сценаріями. Дані були структуровані та зображені у вигляді таблиць.

Базуючись на поставлених вимогах було проведено аналіз можливості застосування наявних технологій та платформ для реалізації спілкування з базою, проксі-серверу, автентифікації. Також були проаналізовані допоміжні бібліотеки для написання додатку, визначено та обґрунтовано доцільність їх використання.

Список використаної літератури

1. iOS [Електронний ресурс] – Режим доступу до ресурсу:
<https://en.wikipedia.org/wiki/IOS>.
2. Android (operating system) [Електронний ресурс] – Режим доступу до ресурсу: [https://en.wikipedia.org/wiki/Android_\(operating_system\)](https://en.wikipedia.org/wiki/Android_(operating_system)).
3. React [Електронний ресурс] - Режим доступу до ресурсу:
<https://reactjs.org/>
4. Passport [Електронний ресурс] - Режим доступу до ресурсу:
<http://www.passportjs.org/>
5. Tripadvisor [Електронний ресурс] - Режим доступу до ресурсу:
<https://www.tripadvisor.ru/>
6. Yelp [Електронний ресурс] - Режим доступу до ресурсу:
<https://www.yelp.com/>
7. Foursquare [Електронний ресурс] - Режим доступу до ресурсу:
<https://uk.wikipedia.org/wiki/Foursquare>
8. Redux [Електронний ресурс] - Режим доступу до ресурсу:
<https://redux.js.org/>
9. Node [Електронний ресурс] - Режим доступу до ресурсу:
<https://nodejs.org/uk/>
10. MongoDB [Електронний ресурс] - Режим доступу до ресурсу:
<https://www.mongodb.org> - 2019
11. Кайл Бэнкер. MongoDB в действии. Москва: издательство “ДМК Пресс”, 2012.

- 12.ExpressJs - [Електронний ресурс]: (Стаття) - Режим доступу до ресурсу: <http://expressjs.com> – 2019
- 13.Ethan Brown. Web Development with Node and Express. US: O'Reilly Media, 2014.
- 14.Bcrypt [Електронний ресурс]: (Стаття) / Wikipedia. Режим доступу: <https://uk.wikipedia.org/wiki/Bcrypt>.
- 15.AWS [Електронний ресурс]: (Стаття) / Wikipedia. Режим доступу: https://uk.wikipedia.org/wiki/amazon_web_services.
- 16.Прамодекumar Дж. Садападж, Мартін Фаулер. NoSQL. Нова методологія розробки нереляційних баз даних. Москва: видавництво "Вільямс", 2013.
- 17.Nginx [Електронний ресурс]: (Стаття) / Wikipedia. Режим доступу: <https://uk.wikipedia.org/wiki/nginx>.